Article

Formal Verification of Authentication and Service Authorization Protocols in 5G-Enabled Device-to-Device Communications Using ProVerif

Ed Kamya Kiyemba Edris [1], Mahdi Aiash [1] and Jonathan Loo [2,*]

[1] School of Science and Technology, Department of Computer Science, Middlesex University, London NW4 4BT, UK; ee351@live.mdx.ac.uk (E.K.K.E.); m.aiash@mdx.ac.uk (M.A.)
[2] School of Computing and Engineering, University of West London, London W5 5RF, UK
* Correspondence: jonathan.loo@uwl.ac.uk

Abstract: Device-to-Device (D2D) communications will be used as an underlay technology in the Fifth Generation mobile network (5G), which will make network services of multiple Service Providers (SP) available anywhere. The end users will be allowed to access and share services using their User Equipments (UEs), and thus they will require seamless and secured connectivity. At the same time, Mobile Network Operators (MNOs) will use the UE to offload traffic and push contents closer to users relying on D2D communications network. This raises security concerns at different levels of the system architecture and highlights the need for robust authentication and authorization mechanisms to provide secure services access and sharing between D2D users. Therefore, this paper proposes a D2D level security solution that comprises two security protocols, namely, the D2D Service security (DDSec) and the D2D Attributes and Capability security (DDACap) protocols, to provide security for access, caching and sharing data in network-assisted and non-network-assisted D2D communications scenarios. The proposed solution applies Identity-based Encryption (IBE), Elliptic Curve Integrated Encryption Scheme (ECIES) and access control mechanisms for authentication and authorization procedures. We formally verified the proposed protocols using ProVerif and applied pi calculus. We also conducted a security analysis of the proposed protocols.

Keywords: 5G; D2D; security protocol; authentication; authorization; formal methods; ProVerif

1. Introduction

In the Fifth Generation mobile network (5G), the end user with their User Equipment (UE) will be able to access network services from multiple resources via new generation Radio Access Network (ngRAN) [1], supported by various technologies. They will also be able to cache and share the accessed services such content data with other users. There is an immense increase in mobile usage and multimedia applications due to end users demanding quick access to services as they are more interested in downloading, caching and sharing content [2]. 5G promises to provide ultra-reliable low latency communications and high availability to support mission critical services, tactile Internet and autonomous transportation to improve users' quality of experience and services [3–6].

5G also intends to solve limitations for peer to peer (P2P) services, as well as offloading the data traffic from backhaul to fronthaul using P2P network such as Device-to-Device (D2D) communications to ease the traffic burden on the 5G core network (5GC) [2]. As an underlay technology, D2D communications will support 5G in enabling new applications and service delivery. This will allow users to easily access proximity services (ProSe) such as content sharing, live gaming, video streaming and offloading traffic to the edge network [7]. In addition, D2D communications will support emergency services, vehicle to vehicle (V2V) communication and the Internet of Things (IoT) [2]. The end users will be involved in content-based operations through D2D communications, which will enhance

user experience by pushing contents to the edge closer to the users [8]. The UE will act as a data consumer as well as a helper in content distribution and delivery [9]. 5G will make network services available anywhere from multiple Service Providers (SPs), and end users will be allowed to access and share the services. This raises new security concerns in the forms of securing data access and sharing that need to be addressed.

One of the challenges for Mobile Network Operators (MNOs) is to provide security and session continuity as users access different networks and third-party services. There is a lack of an efficient multi-purpose mechanism that addresses the authentication and authorization of D2D users in different coverage scenarios to access services from SPs. Moreover, there is a need of security mechanisms that allow the user to cache, share data and delegate their access right to other users. These security solutions must consider security at the network, service and D2D levels in 5G [2]. Federated authentication and authorization can be used to provide flexible security management, accurate tracking of relevant UE data and seamless connectivity [10,11], whereby the MNO/SP can delegate some security and access management to a third-party or the UE [10].

5G non-standalone networks are being deployed, while 5G standardization is now on phase 2 Rel-17 [12], but D2D communications was never standardized for any previous mobile network generations, and it is still the case. However, the legacy system specified some ProSe functionalities and also being considered in many 5G applications [7,9,13–15]. D2D security has been investigated in related work (e.g., [16–19]), but these research efforts did not consider 5G's new use cases such as multiple shareholders, tactile Internet, edge and third-party services [2]. This implies that existing protocols cannot be trusted to address rising security challenges based on the new use cases, such as securing data access, caching and sharing, as well as authorization and user's access right delegation for D2D users in 5G. Therefore, we propose a D2D level security solution, consisting of D2D Service security (DDSec) and D2D Attributes and Capability security (DDACap) protocols that provide authentication and authorization in network- and non-network-assisted scenarios, respectively, enabling two UEs in proximity to access, cache and share data in different scenarios. In this paper, service, data, data object and content are used interchangeably.

Our main contributions are summarized as follows:

- We explore how D2D communications can be used to offload traffic and push contents to the edge closer to the end user in 5G.
- We propose a D2D security solution for authentication, data caching and sharing authorization in network-assisted and non-network-assisted D2D communications scenarios. The solution consists of two security protocols for providing authentication and authorization to D2D users facilitated by Identity-based Encryption (IBE), Elliptic Curve Integrated Encryption Scheme (ECIES) and access control mechanisms.
- We model and formally analyze the proposed protocols using formal methods and automated protocol verifier ProVerif with applied pi calculus.
- We also analyze the protocols security properties conscientiously using two taxonomies.

The rest of the paper is structured as follows. Related works on D2D security in legacy and 5G systems are discussed in Section 2. Section 3 discusses D2D service delivery, system architecture and D2D communications processes. In Section 4, the problem is defined and the proposed D2D security solution is presented. The modeling of the proposed security protocols is presented in Section 5. In Section 6, the formal verification of the protocols is presented. Section 7 analyzes and evaluates the protocols' security properties. The paper is finally concluded in Section 8.

2. Related Work

5G will provide enhanced Mobile Broadband, ultra-reliable low latency communications, massive machine-type communications to support augmented reality, virtual reality, eHealth systems and smart cities [2,6]. With D2D communications used as an underlay technology in 5G, its legacy security solutions for Long-Term Evolution (LTE) will be inad-

equate to address 5G-enabled D2D communications. Zhang et al. [16] proposed a secure data sharing protocol for a UE to access SP services in Long-Term Evolution Advanced (LTE-Advanced) network. It provided mutual authentication, non-repudiation, traceability, data confidentiality and integrity. In [17], the authors investigated security and privacy for D2D communications using LTE-D2D architecture and applications and proposed security solutions to address application and physical layers. This work was based on Fourth Generation (4G) architectures and old D2D communications technologies.

In [19], the authors explored key distribution in D2D communications and proposed a lightweight key distribution scheme using acceleration sensors, introducing a lightweight extreme points extraction and filtering algorithm to extract extreme points for key generation. In [20], the authors proposed a universal authentication and key agreement protocol for D2D communications, providing privacy-preserving, session key generation and mutual authentication between D2D users using different scenarios. Wang et al. [21] proposed an anonymous authentication and key agreement secure protocol for D2D communications, providing mutual authentication, session key, real identity leakage prevention and secure communications. However, service access authorization and data security were not covered.

In [22], the authors introduced a new privacy-preserving security architecture for a fog computing model with the cooperative D2D communication support and proposed three lightweight anonymous authentication protocols to support three distinct circumstances in D2D-Aided fog computing to secure IoT applications. Seok et al. [23] explored the security issues raised as a result of emerging IoT technology with 5G. They then proposed secure D2D communication based on Elliptic Curve Cryptography (ECC) and lightweight authenticated encryption with associated data ciphers to address IoT devices' security. It mainly covered IoT security issues when emerged with D2D communications, not focusing on D2D specific scenarios or service authorization.

Wang et al. [24] discussed 5G-enabled Vehicular ad hoc networks (VANETs) security related issues and then proposed a hybrid D2D message authentication scheme for D2D-VANETs, providing mutual authentication for vehicle-to-vehicle communication. In [25], performance and security issues e-health IoT devices are explored and a new mutual authentication protocol for m-health systems based on D2D communication to protect data shared between IoT devices is proposed. Wang et al. [26] investigated D2D group communication security and proposed a dynamic group key agreement protocol to provide a constant-round authentication in D2D group communications, whereby the D2D users communicate and authenticate without NodeB or base station.

In [18], the authors explored authentication and key management for D2D/5G communications and proposed a framework based on the physical layer that provides entity authentication and key establishment solution to address the related issues. Wang et al. [27] also investigated physical security for D2D communications. Then, they proposed an access selection scheme for D2D users to protect cellular users against eavesdropping, using jamming techniques to interrupt the eavesdropper, and an optimization method to prove optimal threshold and developed an iterative algorithm to determine the optimal threshold. Li et al. [28] explored communication delay and network capacity issues. They proposed a security and energy-aware collaborative task offloading for D2D communications. They formulated the collaborative task offloading problem that minimizes the time-average delay and energy consumption of devices while ensuring data security.

Yan et al. [29] discussed the importance of securing communication data and thus proposed a security scheme based on trust and evaluated D2D users for D2D communications data access control in LTE based on Attribute-Based Encryption (ABE) mechanism. In [30], data security, identity privacy and system scalability are explored. The authors proposed a robust and scalable data access control scheme in D2D communications with scalable system attributes and each base station controls the attribute universe individually. The core network server acts as the central authority, and the complicated decryption can be offloaded from an inadequate device to a more capable one.

The discussed related works address some of the security issues that are affecting different D2D communications applications providing mutual authentication, session key generation, access controls and data security. However, they do not cover D2D communications from an abstractive point of view, whereby we explore D2D security at the network and service levels of D2D communications. We address entity authentication and access, caching, sharing authorization, data security and user's access right delegation for D2D users in 5G. Most of the related works address authentication or service access control for D2D communications, but not both. In addition, data dissemination, multiple shareholders and supporting technologies in 5G such as Content Delivery Network (CDN) [31], edge services by third party SP and end users' demand for high reliability and high availability have not been considered. Our proposed solution provides robust authentication and authorization mechanisms that can be integrated with the next generation mobile network. Moreover, in our proposed solution, two security protocols are designed to address security in two D2D communications scenarios. We consider the changes in the mobile system and integration with security mechanisms specified in [1] at network and service levels of the 5G and D2D communications systems [2] as well as address security of the UE, transmission and the data.

3. Service Delivery in 5G-Enabled D2D Communications Network

To distribute and deliver content to the end user efficiently, content delivery models such as CDN must be adopted; cache servers deployed at edge [32] and UEs are used as cache nodes [33] to support content caching. The introduction of CDN and content caching in mobile network can be integrated with Content-Centric Networking (CCN) [34] to turn mobile network into a content aware network [2].

3.1. System Model

5G requires an efficient service system, access and delivery models to support its objectives, which can be achieved by integrating other architectures and services with 5G. We adopt a system model that consists of Cloud Radio Access Network (C-RAN) [35], CCN [34] and D2D communications [7] as suggested in [15]:

- C-RAN is adopted for the centralization of base stations, resources virtualization and edge service deployment.
- CCN is adopted for content distribution and caching.
- D2D communications is adopted for direct communication, content dissemination and data offloading.

C-RAN centralizes the baseband resources to a single baseband unit called Baseband Unit (BBU) pool, connecting to the remote radio heads transceivers combing the function of the base station. In addition, the interface between BBU and radio heads has been changed from circuit fronthaul to packet fronthaul. BBU is divided into the distributed unit responsible for physical layer and real-time MAC layer process and the centralized unit responsible for upper layer computations process [35]. The UE connects to ngRAN, Security Anchor Function (SEAF) or Access and Mobility Function (AMF) to access the network. The 5GC consists of various network functions such as Session Management Function (SMF), Authentication Server Function (AUSF) and Unified Data Management (UDM) [36] residing in the Home Network (HN).

The system model in Figure 1 consists of the following entities:

- UE: Transmitter and receiver.
- Next generation NodeB (gNB): Base station connecting UE to the network.
- HN: UE registers to and receives services from it.

The CCN protocol can be embedded into the UE, BBU pool, edge routers and 5GC [15], or the control and user plane enhancement can be implemented to enable services such as CCN as 5GC network functions and extend the interfaces to support CCN-protocol data unit sessions [37]. To access services, the UE must be registered with MNO/SP and subscribed to those

services, whereby the UE can request to access the network, perform primary authentication with HN and then request to access services depending on the service level agreement [10].

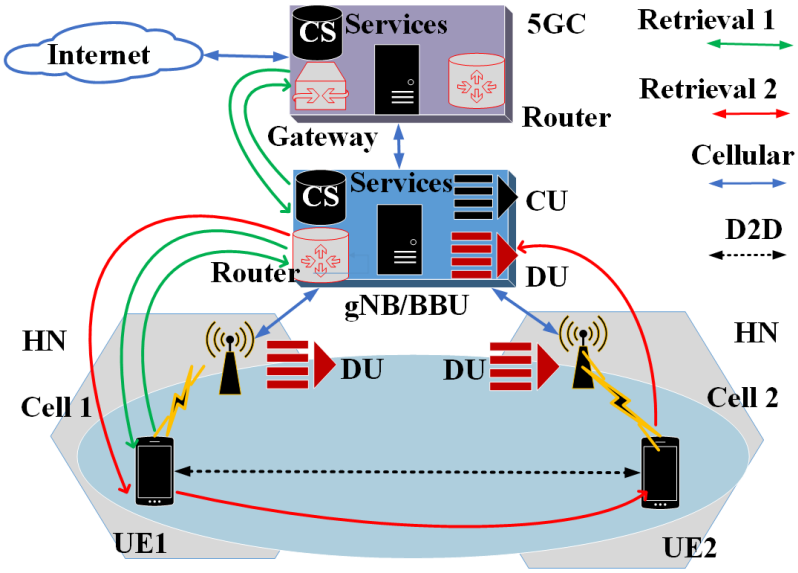

Figure 1. System model.

After a successful primary authentication, a secondary authentication and service authorization may be required between the UE and SP, to authorize the UE to access third party services, cache and share data with other UEs [38,39]. MNO/SP is responsible for services subscription, authorizing service access and content retrieval. MNO controls the user's access to the network, connection establishment, resource allocation and security management. MNO/SP might also want to hide their visibility or deny the UE from accessing their services. There is an inter-operator agreement between different networks to provide inter-operator roaming services.

3.2. Service Access and Delivery

With content access and retrieval process, the original named data object is published by the content provider/data owner using a Uniform Resource Identifier (URI) of the data as the data identity (ID) or data name which is advertised on the MNO/SP network. Thus, for service discovery, the UE searches for specific data by data ID and some contact information from the owner's routing table are requested, which are in search tolerance of the data ID [40]. The UE must learn the ID/key mapping that associates a UE's ID with the data to be able to request data objects on the network. Additionally, the UE which has been granted access to the data can publish the data based on the content owner and SP security policies. The named data object and UE's ID are mapped together into logical/node address. The UE interested in the data uses the logical address to send interest messages to the UE in possession of the data.

The process of content dissemination and service delivery starts with the UE generating an interest message which is forwarded to BBU through radio heads. BBU starts a CCN forwarding process to match the request content name [15,37]. If the targeted content exists in BBU pool content store, the content is sent back to the UE, fulfilling the initial request. During content forwarding, the involved entities can choose to cache a content replica or not, which might also depend on the security parameters such as cache authorization. The content retrieval process of D2D communications involves cache satisfaction at BBU, UE and 5GC. Two applications of content retrieval are considered: (1) where the content is being matched at the 5GC; and (2) where the content is being matched locally by another UE in proximity, as shown in Figure 1.

However, if the content request is not matched at the UE locally or from the UE in coverage or BBU, the interest is forwarded to 5GC. If it is matched in the content store of 5GC, then the content is pushed backwards to the UE. The traffic burden is reduced on backhaul if the request does not go to the 5GC and the content retrieval is achieved by using fronthaul and D2D links. BBU pool discovers cached content by associated devices and content transmission is performed using D2D communications. To achieve this process, the security requirements for accessing network and service, data caching and sharing must be achieved. We assume that initial authentication of each UE with its HN has been performed and the two HNs have already agreed on the roaming service agreement. Before the establishment of secure D2D communications, each UE sends a verification request to its gNB or another UE. Then, the authentication and authorization procedure is initiated, the UEs exchange security vectors and they verify each other using the proposed protocols for a secure data transmission.

D2D discovery and communication are performed during content dissemination. Content discovery is based on CCN interest and data object messages, which is associated with the data name and UE's ID.

3.3. D2D Communications Process

The D2D communications process consists of device discovery, link step up and data transmission stages similar to that defined for ProSe in [7] to satisfy a service request and establish a communication session. The communication of the UE in cellular coverage and initiation of direct communication between two UEs are controlled by the base station, whereas the complete direct communication and out of coverage are controlled by the D2D devices. The requesting UE can send a broadcast message to discover any UE in proximity or a specific content and the UE with the content can send a broadcast about its data. If a UE receives a request message, it sends a response message via link setup stage. The data transmission can be established with or without the help of gNB. The full security process is presented in Section 5.

3.3.1. D2D Communications Network Assisted

D2D connectivity is initiated by gNB via SMF/AMF. When data packets reach the SMF, data packets in the form of Internet Protocol (IP) or CCN can be analyzed by the 5GC whether the link can be considered as a D2D link or not. This process starts from Step 1. It is assumed that a UE can discover other UEs in proximity, hence a UE can also initiate D2D connectivity and start the process from Step 3.

The generic and concise D2D connectivity process in Figure 2 is explained in the following steps:

1. When the SMF receives a session request message, it checks if the transmitter and receiver are within the same cell and the channel condition can support D2D communications. Then, it initiates the D2D connectivity with the help of gNB. The request messages include UE IDs and data name.
2. Then, gNB starts a strategy list of communication patterns, push uplink shared channel, physical uplink control channel, power indicator, scan spectrum and time. After analyzing the request messages from SMF, it sends the message to UE_1.
3. UE_1, hoping to make D2D connectivity with UE_2, sends regular information in the physical radio access channel with binary code, UE ID, data name and the UE can also detect received information from other UEs.
4. After matching the targeted UE_2, the initiating UE_1 sends the D2D connectivity request to gNB.
5. Then, gNB analyzes the D2D connectivity request and chooses the best communication pattern for the D2D pair based on the conditions of the channel and cell resource utilization.

6. Sometimes the D2D connectivity request fails; the normal cellular communication mode is adapted and the communication between the UEs goes back to the conventional cellular mode.
7. Conversely, if the D2D mode request including orthogonal or reused pattern is permitted, gNB establishes a strategy list including communication patterns, power indicator, scan spectrum and scan time and sends it to UE_1.
8. After completing the channel measurement such as normalized interference intervals, UE_1 provides an update to gNB.
9. gNB allocates spectrum resource to UE_1 and lets UE_2 recognize the same channel. Training sequences at the allocated channel are sent by the UE_1 to help UE_2 to get the link quality. Then, UE_2 sends a confirmation message back to UE_1 after checking that the link quality is suitable and UE_1 confirms it to gNB.

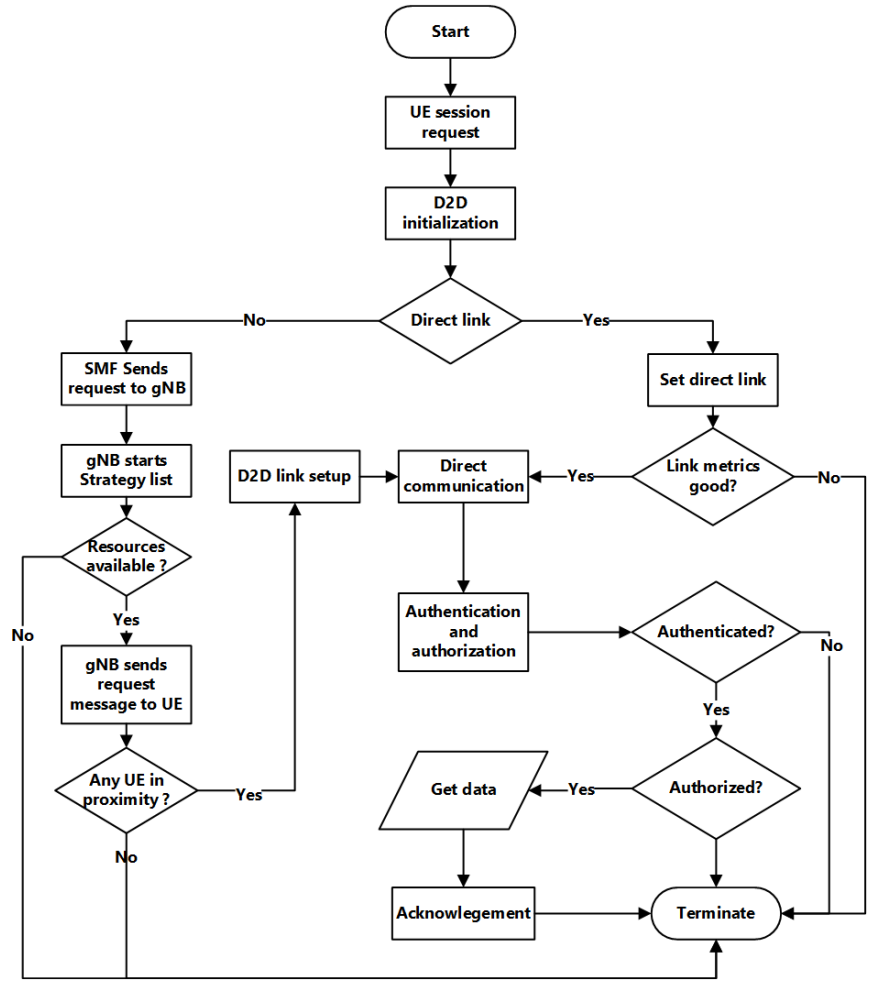

Figure 2. D2D connectivity process flowchart.

3.3.2. D2D Communications Non-Network Assisted

This is similar to Mobile Ad Hoc Network (MANET); however, the D2D link uses a reserved cellular licensed spectrum for an out-band communication, also referred to as public safety network. This is a direct communication between UEs with a controlled link establishment by the UEs.

1. UE_1, hoping to make direct communication, transmits beacon signals with low frame rate to reduce signaling overhead.
2. UE_2, in proximity, responds by sending acknowledgement messages.
3. The received messages are evaluated based on metrics, such as Signal-to-Interference-plus-Noise Ratio (SINR).

4. The UEs can start direct communication with each other, when a received signal is above the predetermined metrics threshold with a good quality of service.

The beacon signals are transmitted over channels that are dedicated for transmitting control signals, whereas data exchange is performed using shared data channels. To establish safe and secured links in D2D communications, the beacon signals broadcast contains security vectors within the packet frames. 5G UEs are capable of an authentication process with low signaling overhead.

4. Security in 5G-Enabled D2D Communications Network

This section presents the problem definition and the proposed security solution for D2D communications in a 5G system.

4.1. Problem Definition

5G promises to create new use cases, support vertical services, reduce backhaul traffic and push content closer to the users facilitated by D2D communications. However, security in D2D communications has been a serious concern due its wireless characteristics [41], omission of third-party and 5G's heterogeneous nature, which present several security vulnerabilities that put the whole network at risk from new and old threats [2]. D2D-based content distribution [9,42] and traffic offloading will increase the vulnerabilities of entities and data. In addition, security for services and user's data at the edge is another concern.

Security issues in D2D communications and CCN systems [43–46] have been investigated separately, but not for an integrated system. The system can be affected by attacks such as eavesdropping, data fabrication, spoofing, content poisoning, privacy violation, denial of services, masquerading and linkability attacks [47–49]. Due to the self-management of D2D UEs, implementing security and privacy is difficult and more challenging in 5G as privacy violation increases when D2D users are accessing ProSe. The HN and visited network could also be curious and eavesdrop on D2D communications. These attacks require robust authentication methods for a secure communication between D2D users [16] and access control methods to secure service access and delivery [29,30].

After primary authentication methods such as 5G-Authentication and Key Agreement (AKA) [1] is used to authenticate the UE to the HN [50], the UE is usually granted static authorization, but, with edge computing, third-party services, increasing demand for multimedia contents and tactile Internet in 5G, access authorization is more complicated. Therefore, multi-level security measures must be used to allow interoperability between different security domains. MNO/SP should be able to restrict access to its services even when the UE is not in coverage. Thus, network access and service protocols must be used to allow the UE gain access to services provided by MNO/SP in HN or data network [39,50] and enable secure service sharing between UEs.

4.2. Proposed D2D Security Solution

After gaining access to network and services, the UE can request other UEs in proximity to share services following D2D communications process as specified in [13]. We propose a D2D level security solution to address the mentioned security issues to enable UEs access, cache and share content securely in different D2D communications scenarios of 5G such as roaming, non-roaming, in coverage and out of coverage shown in Figure 3 and explained as follows:

1. Scenario A: Intra-operator, non-roaming and same cell, both UEs are in cellular coverage of gNB, same HN and subscribed to the same MNO .
2. Scenario B: Intra-operator, non-roaming and different cells, both in HN coverage, subscribed and served by the same operator but in different cells.
3. Scenario C: Inter-operators and roaming, one UE is in HN, while the other UE is roaming in visited network, and both users subscribe to different operators.

4. Scenario D: Inter-operators and out of coverage, both UEs are out of coverage and can share content without involving their HNs. It also applies to emergency situations. Both UEs subscribed to different operators.

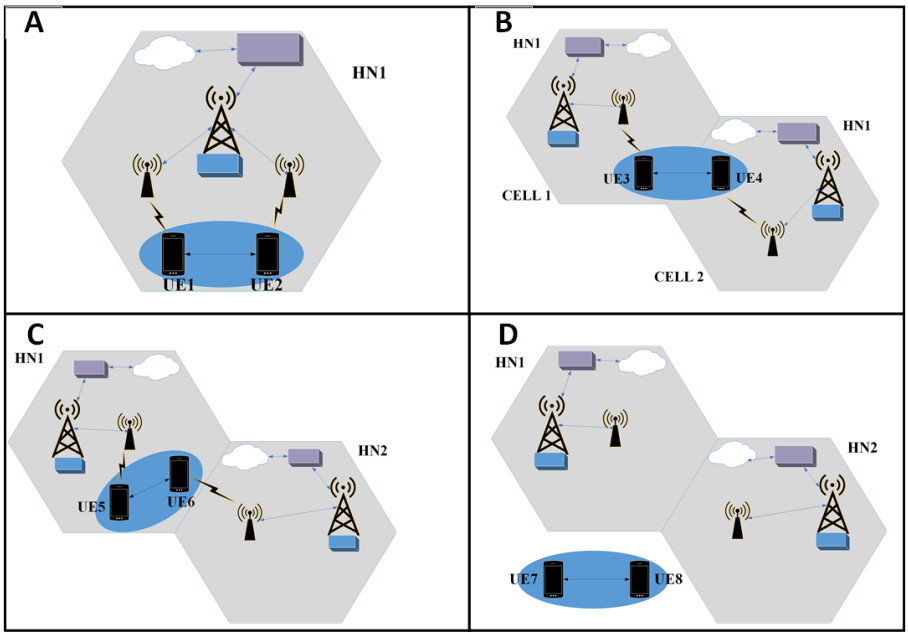

Figure 3. D2D communications scenarios. (**A**) same cell/same operator, (**B**) different cells/same operator, (**C**) roaming mode/different operators and (**D**) out of coverage/different operators.

The D2D security solution consists of DDSec and DDACap protocols, which intend to provide authentication and authorization to access, cache and share data between two UEs in different D2D scenarios. Additionally, a UE will be able to transfer its access permissions and abilities to another UE. DDSec is intended for network-assisted D2D communications, while DDACap is intended for non-network-assisted direct D2D communications.

The UE_i is free to choose any publicly known information as its ID such as public key or IP address based on IBE scheme [51]; this can be used together with another form of ID such as the Generic Public Subscription Identifier (GPSI) from primary authentication [1,50] and external ID (EID) and federated ID (FID) from secondary authentication and authorization procedures [38,39], respectively. This eliminates the need for distributing keys as in the traditional AKA or Public Key Infrastructure (PKI) and use of Subscriber's Permanent Identifier (SUPI) outside the HN as specified in 5G standard [1]. For integrity, privacy and uniqueness, the UE uses a pseudonym ID in D2D communications calculated from the UE's preferred choice of ID, $hash(x(ID))$, where x is the public key of UE, randomly generated using hash function $hash(PK_{UE_i}(ID))$ [40]. The relation between UE and data is determined by pseudonym IDs of the UE and the data which become the logical address $nAdd$ of the UE with the data. The data's unique ID $D1$ is generated by hashing the data name $hash(URI)$. The UE_i would first hash both the UE's pseudonym ID and $D1$ as $hash(UEID, D1)$ and then publish it as its logical address. The keys and IDs are generated using ECIES [52] according to 5G standard [1].

4.3. Security Requirement

The UE is capable of cryptographic computation using its Universal Subscriber's Identity Module (USIM); each UE generates its own self-certifying public key PK_{UE_i} and private key SK_{UE_i} [53]. The $PK_{(UE_i)}$ is used as part of its pseudonym ID. The UE_i generates the secret key SK_{UE_i} by selecting a random number using chosen binary ECC, and then its public key PK_{UE_i} is computed from SK_{UEi} [52]. Each UE_i sends its public key and pseudonym ID to gNB in terms of network-assisted application and to each other in direct communication application. Traditionally, certificates would be used to bind the UE_i to

its public key, however in this case self-certifying public keys are used. The UE_i stores its public/private keys PK_{UEi}, SK_{UE_i} and public key hash $hash(PK_{UE_i})$ while gNB stores the public keys, pseudonym ID and other relevant information of all users in the cell. A timestamp is used to prevent replay attacks and check the validity of the token. It is evaluated by checking if it is within a given time-window relying on the clock of the UE. The message timestamp must be equal or slightly off by seconds to the current time/date, while the token lifetime time/date must be after the current time/date. Furthermore, the UEs can generate hash functions, symmetric and asymmetric keys and digital signatures achieved using ECIES and IBE. ECIES is suitable for systems with low computational capabilities and resources, such as mobile devices and smart cards due to the point multiplication operation. The proposed protocols use ECIES as applied by 5G to conceal SUPI and generation of the anchor key [1].

The expected security properties to be achieved by the DDSec and DDACap protocols are informally defined before the formalization of the protocol properties. The taxonomies in [54,55] are adopted for precise formal analysis referred to as Set 1 and Set 2, respectively. In Set 1, the security properties are specified from an Agent A's point of view, with four levels defined between two Agents A and B: aliveness, weak agreement, non-injective agreement and injective agreement. In Set 2, the security protocol should meet the following security properties: mutual entity authentication, mutual key authentication, mutual key confirmation, key freshness, unknown-key share and key compromise impersonation resilience.

4.4. Authentication and Authorization

The UE_i authenticates to the HN with SUPI as part of the primary authentication. After a successful authentication, it generates its self-certifying key pairs and pseudonym ID and sends a service request with pseudonym ID to gNB for D2D services, where gNB maintains a registration table to associate the $UEIDs$ with PK_{UE_i} and service information. For example, UE_A with data can advertise or publish its ID UA and $DataName$ or UE_B interested in the data can send an interest its ID UE_B and $DataName$ using UE_A's logical address $nAdd$. Then, mutual authentication is initiated and achieved using PK_{UE_i}, $hash(PK_{UE_i})$ and nonce. After authentication, UE_A sends data to UE_B via conventional routing mechanisms of D2D communications and CCN processes. For routing, the UE can reach another UE in proximity using the UE's $nAdd$ and each UE in the routing path without exposing any other interface. If the UEs did not communicate before, then they have no cached security context. If they have communicated before, then the valid cached data, such as IDs and public keys, can still be used for authentication and authorization.

The UEs can get authorized to access, cache and share the data without involving BBU pool or gNB as central authority, even though the security context such as data encryption keys might be set by third party. Therefore, authorization procedure might depend on the MNO set policies and service agreements between the UE and HN or SPs such as access controls, FID and security tokens. It leverages on security tokens, user access rights delegation, capabilities and attributes of the user and the data [39,56]. The UE can also use GPSI or FID as its global profile to the UEs previously authorized via a federated-based authorization instead of the pseudonym ID [39].

The data owner creates capabilities in the form of labels for the UE and data object during service authorization by SP, and each object is given a specific set of access rights that can be granted to a UE [39]. The capabilities are represented as a dot-separated sequence of numbers, a string $.i_1.i_2...i_n$ for some value n where $i_1, i_2, ..., i_n$ are integers [56]. The user's capabilities $ucap$ are transferable through delegation process [39]. The data object consists of the data object name, metadata, payload and other non-encrypted configuration. The metadata consists of publication date, expiry date, creation time and access control profile. The access attribute and capability are specified in access control profile which includes data's capabilities $dcap$, a symmetric nonce key K_{d1} and its key ID $(hKd1)$ for decrypting the data. The K_{d1} is the $hash(nonce, DataName)$ while the key ID is the hash of the key

$hash(K_{d1})$ signed by the data owner's private key SK_{DO}. For D2D communications, the UE with data and delegated attributes creates an $ACap$ token consisting of authorization attributes and capabilities, $ACap = \{D1, DA, hKd1, exp\}$:

- $D1$: DataName.
- DA: Dataset delegation (true), label $(ucap2.3)$, data owner ID $(hDOID)$, data owner's public key PK_{DO}, token $ACap$ owner ID (UB), scope (offline access, cache, share, any).
- hKd1: Nonce key ID signed by data owner's private key SK_{DO}
- Exp: Token expiry date.

The delegated access rights include dataset delegation; if set to true, then it permits the delegation of attributes and capabilities to other users in line with the data owner's set permissions. It also includes the scope of user/client's access rights in relations with the data object, data owner's ID ($hDOID$), data owner's public key and user's ID of the UE in possession of data. The nonce key is a symmetric key for decrypting the data object. The $ucap$'s 2.3 should match the $dcap$ 2.3, i.e., $2 = caching$ and $3 = sharing$. For the D2D users, the access duration depends on the expiry date of the data not the $ACap$ expiry date. The $ucap$ is not user specific; anyone could access the data so long as they are in possession of the right $ucap$ and encryption key as set by the data owner. In a scenario where the UE wants to share its owned data, it can generate an encryption key and $ACap$ token, hence becoming a data owner and a token generator.

5. Modeling of the Proposed Protocols

This section presents the modeling of the proposed DDSec and DDACap security protocols. The UEs should be able to access and share restricted services even in the case where certain roaming agreements do not apply. At this security level, the UE can use some of the primary, secondary and federated security context such as preshared session keys and federated IDs, however it is still capable of generating its own cryptographic primitives and security vectors, as explained in Section 4. It leverages on security parameters and guarantees used in network access security [50] and service level security [39] mechanisms. These protocols apply cryptographic primitives such as those specified in 5G standard [1] to achieve the security requirements of 5G-enabled D2D communications. The cryptographic primitives used include symmetric and asymmetric encryption, one-way hash function, digital signature and message authentication code.

5.1. DDSec Protocol

The proposed DDSec protocol is modeled using UE_A, UE_B and gNB entities based on the system model presented in Section 3 and a network-assisted D2D scenario. For simplification, it is assumed that UE_A has already been granted access to the service (data) and has been authorized to cache and share data. Therefore, UE_A can share data and delegate its access right and capabilities to UE_B. UE_A and UE_B can use the proposed DDSec protocol to mutually authenticate and share data assisted by gNB.

DDSec Protocol Message Exchange and Execution

Before running the protocol, UE_A and UE_B would have achieved a successful network access authentication through primary authentication procedure using 5G-AKA/5G EAP-AKA' protocols [50] with the HN. The communication between the UEs and gNB is secured with respective keys K_{gNBa} and K_{gNBb}, derived after a successful primary authentication. UE_A is also in possession of a symmetric key K_{d1} for decrypting the data, received as part of the service authorization procedure that grants the UE access to the service as presented in [39]. UE_A can advertise to other devices in proximity including gNB about the data in its possession; in addition, gNB would have information on IDs, previous data transmissions, data names and data owners. Additionally, when another UE sends an interest message about that data, gNB assists in the D2D device discovery and link setup between the two UEs. The protocol messages exchange is among UE_A, UE_B and gNB entities for Scenarios

A–C in Figure 3 Section 4.2. In the case of Scenarios B and C, two gNBs would be used. Figure 4 illustrates messages execution for Scenario A, with reference to notation in Table 1 and described as follows.

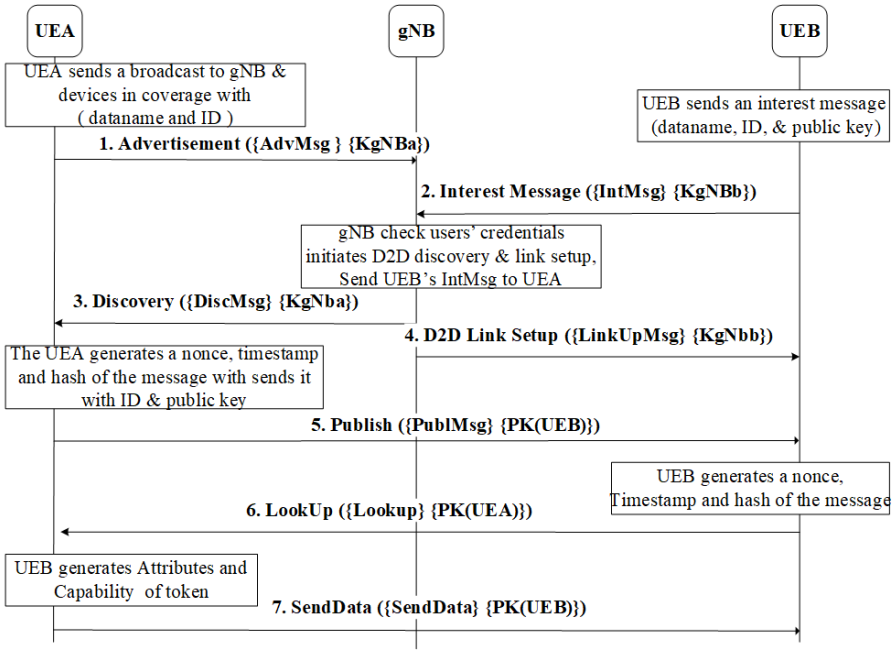

Figure 4. DDSec protocol message exchange flow.

Table 1. DDSec and DDACap protocols notation and description.

Notation	Description
$UE_A/UE_B/UE_C$	UE
gNB	new generation NodeB
Rand_a/Rand_b/Rand_rb/Rand_rc	Nonce
PK(X)	public key
SK(X)	private Key
Exp	expiry date
URI	DataName
Ts	time stamp
Acap	security token
$nAdd$	logical/node address
D1	hash (DataName)
K_{gNB}	session key (UE and gNB)
hKd1	nonce key ID
UA/UB/UC	pseudonym ID hash(X)UEID))
DOID	data owner's ID
hDOID	hash(DOID)
h(x)	hash value of message x
{x}{k}	message encrypted with key K

`Msg1 UEA` $\rightarrow$ `gNB:({AdvMsg},{KgNBa})`

UE_A sends an advertisement to gNB that it is in possession of data. The advertisement message $AdvMsg$ includes the data name $DataName$, its pseudonym ID (UA), logical address $nAdd$ and public key PK_{UE_A} encrypted with preshared key K_{gNBa} with gNB. Then, gNB updates its registration table and BBU Pool updates the intra cell content database with the received message.

`Msg2 UEB` $\rightarrow$ `gNB:({IntMsg},{KgNBb})`

UE_B sends an interest message $IntMsg$ to the affiliated BBU pool via gNB, which includes

its pseudonym ID (UB), data name $DataName$ of the data it is interested in and its public key PK_{UE_B} encrypted with preshared K_{gNBb} with gNB.

`Msg3.gNB→UEA:({DiscMsg},{KgNBa})`

gNB with BBU pool check intra cell content database for an interest match within the local cell coverage or the D2D users in proximity via AMF which may have information about other UEs from other cells in intra or inter operator scenarios. If there is a match, gNB initiates the D2D communications process of discovery and link setup between UE_A and UE_B via D2D communications supported by SMF. In this case, BBU pool finds a match from UE_A, gNB sets up a link with UE_B and forwards (UB), $DataName$ and PK_{UE_B} to UE_A in $DiscMsg$ message encrypted with key K_{gNBa}.

`Msg4.gNB→UEB:({LinkUpMsg},{KgNBb})`

gNB forwards UE_A's (UA), data name $DataName$, logical address $nAdd$ and public key PK_{UE_A} to UE_B in $LinkUPMsg$ message encrypted with key K_{gNBb}.

`Msg5.UEA→UEB:({PublMsg},{PKUEB})`

After the confirmation of link quality of the allocated channel, UE_A initiates the authentication request by sending a $PublMsg$ message that includes its pseudonym ID (UA), data name $DataName$, data ID ($D1$), a nonce $Rand\_a$ as authentication challenge, a timestamp Ts_1 and hash of the message $hash(UA, DataName, D1, Rand\_a, Ts_1)$ signed with its private key SK_{UE_A}. and $PublMsg$ is encrypted with UE_B's public key PK_{UE_B}. The hashing is used for integrity and message authentication and timestamp for replay attack protection.

`Msg6.UEB→UEA:({LookUp},{PKUEA})`

When UE_B receives Message 5, it responds with a $LookUp$ message that includes its pseudonym ID (UB) and data name pseudonym ID $D1$, confirming the data name and its ID. UE_B also returns the nonce $Rand\_a$, together with new nonce $Rand\_b$, a timestamp Ts_2, hash of the whole message $(UB, D1, Rand\_a, Rand\_b, Ts_2)$ signed with its private key SK_{UE_B} and encrypt the $LookUp$ message with UE_A's public key PK_{UE_A}. UE_B authenticates UE_A by sending back $Rand\_a$ together with $Rand\_b$.

`Msg7.UEA→UEB:({SendData},{PKUEB})`

When UE_A receives Message 6, it generates attribute and capabilities token $Acap$ using the permission delegation authorization granted by original data owner. Then, UE_A returns $Rand\_b$ to confirm a successful authentication of UE_B together with the requested $DATA$, $Kd1$ symmetric key to decrypt the data, $ACap$ token authorizing UE_B to cache, share data and delegate its access rights to another UE and hash of the symmetric key and the token $hash(Kd1, ACap)$ signed with SK_{UE_A} in $SendData$ message encrypted with PK_{UE_B}.

5.2. DDACap Protocol

The proposed DDACap protocol is modeled using UE_B and UE_C entities based on the system model presented in Section 3 and a non-network-assisted D2D scenario with the same assumptions and security requirements as DDSec protocol in Section 5.1. The UEs would be allowed to access and share data even in the scenarios where certain roaming agreements do not apply, out of coverage and during a disaster situation. The UEs use some security context from DDSec if it applies to the scenario, however they are still capable of generating their own cryptographic primitives and security parameters such as timestamps and tokens. After UE_A shares the data and delegates its capabilities to UE_B, UE_B can use the similar security attributes and capabilities for user and data object to cache and share the data with UE_C without the assistance of the network. For UE_B to share the data with UE_C, both entities do not have to be on the same security level. UE_B and UE_C can use the proposed DDACap protocol for authentication and authorization of data caching and sharing.

DDACap Protocol Message Exchange and Execution

The overview of the DDACap protocol execution involves UE_B and UE_C. The protocol initiation can be performed by the interested UE or the UE with data. In this case, UE_B with data initiates the direct communication by sending beacon signals with advertisement

message and UE_C in proximity responds with interest messages. When the messages are evaluated and the signal has exceeded the predetermined metric threshold, both UEs start to communicate directly. The control signal and the data are exchanged over share data channels. The protocol messages exchange is between UE_B and UE_C for Scenario D in Figure 3 (Section 4.2), as illustrated in Figure 5, similarly with reference to notations in Table 1 and described as follows.

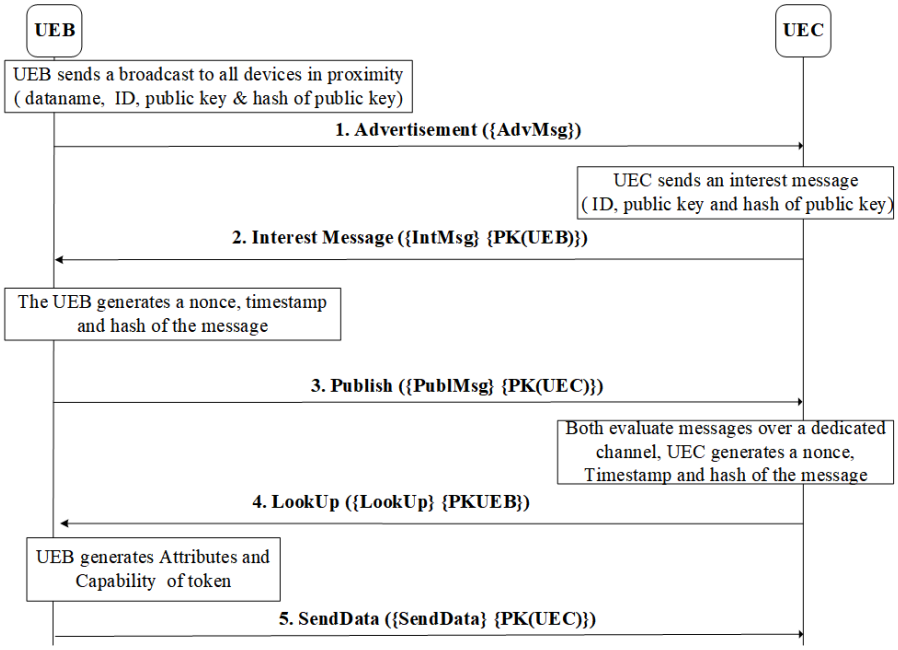

Figure 5. DDACap protocol message exchange flow.

`Msg1 UEB` $\rightarrow$ `UEC:({AdvMsg})`

UE_B sends beacons signals with an advertisement message $AdvMsg$ by broadcasting to the D2D users in proximity which includes UE_C. It includes its pseudonym ID (UB), data name $DataName$, logical address $nAdd$, its public key PK_{UE_B} and hash of its public key $hash(PK_{UE_B})$ signed with its private key SK_{UE_B}, the hash is used for integrity and message authentication.

`Msg2 UEC` $\rightarrow$`UEB:({IntMsg},{PKUEB})`

UE_C sends an interest message $IntMsg$ to UE_B by replying $AdvMsg$. It includes its pseudonym ID (UC), data name $DataName$ of its interest, public key PK_{UE_C} and its hash ($hash(PK_{UE_C})$ signed with its private key SK_{UE_C} and encrypted with UE_B's public key PK_{UE_B} hoping to make a direct communication.

`Msg3.UEB`$\rightarrow$`UEC:({PublMsg},{PKUEC})`

When UE_B receives Message 2, it responds with publish message $PublMsg$. UE_B generates a nonce $Rand\_rb$ as authentication challenges, a timestamp Ts_3 for freshness and replay protection, pseudonym ID of $DataName$ ($D1$) and hash of the whole message $hPublMsg$ =$hash(UB, D1, Rand\_rb, Ts_1)$ signed with its private key SK_{UE_B}. UE_B sends the $PublMsg$ message that includes (UB), $D1$, $Rand\_rb$, Ts_3 and $hPublMsg$ as an authentication request encrypted with UE_C's public key PK_{UE_C}.

`Msg4.UEC`$\rightarrow$`UEB:({LookUp},{PKUEB})`

Both UEs evaluate the received messages to establish direct communication over dedicated channels. Then, UE_C responds with $LookUp$ message that includes its ID (UC), the data name $D1$, new nonce $rand\_rc$, nonce $Rand\_rb$, a timestamp Ts_4 and hash of the whole message $hash(UC, D1, Rand\_rb, Rand\_rc, Ts_4)$ signed by its private key SK_{UE_C}. The $LookUp$ message is encrypted with PK_{UE_B}. UE_C authenticates UE_B by sending back $Rand\_rb$ together $Rand\_rc$.

`Msg5.UEB`$\rightarrow$`UEC:({SendData},{PKUEC})`

When UE_B receives Message 4, it checks if there are any delegated permissions; if there

are, then it generates attributes and capabilities token $Acap1$. It returns $Rand\_rc$ to confirm a successful authentication of UE_C, together with requested $DATA$, $ACap1$ token with abilities and capabilities string, $Kd1$ symmetric key to decrypt the data and hash of the symmetric key and token $hash(ACap, Kd1$ signed with SK_{UE_B} to UE_C in $SendData$ message encrypted with PK_{UE_C}.

6. Verification of the Proposed Protocols

This section formally models and verifies the proposed protocols using formal methods and ProVerif. The security properties are formalized with ProVerif results.

6.1. Formal Methods

Formal methods and automated tools have been used to verify mobile network security to assess their security properties [20,50], hence providing strong security guarantees. Security protocols properties are very challenging for most verification techniques and tools. This is due to the use of cryptographic primitive and algebraic properties that are hard for symbolic reasoning, hence certain tools and manual proof checks are not suitable [50]. However, there are automated verification tools that can be used for protocol modeling and analysis such as Automated Validation of Internet Security Protocols and Applications (AVISPA) [57] and ProVerif [58].

ProVerif [58] is an automatic tool for analyzing security protocols, with Dolev–Yao [59] as the adversarial model and enabling the verification of security properties, by supporting equational theories defined by users. It supports the underlying theory of abstraction, focusing on unbounded sessions, and uses precise Horn clause abstraction and applied pi calculus [60] as a formal language for modeling security protocols. The syntax is paired with a formal semantics to enable reasoning about protocols. It also supports a variety of cryptographic primitives, modeled by equations and rewrite rules. Additionally, it takes the security properties to be proved, such as authentication, secrecy or observational equivalence, as input. It translates the protocol into an internal representation by Horn clauses, which makes crucial abstractions in order to support an unbounded number of sessions. Cryptographic primitives are modeled as functions, messages as terms built over an infinite set of names such as $a, b, c, ...$, variables such as $x, y, z, ...$ and a finite set of function symbols such as $f1, ..., fn$. The syntax of ProVerif process language is shown in Table 2 and a full description can be found in [58]. Therefore, ProVerif suits the type of analysis conducted in this paper. It has been used before to formally verify security protocols [38,39,50].

Table 2. Core language: term and process grammar.

Term	Grammar
M, N ::=	term
a, b, c, k, s	name
x, y, z	variable
h(M1, . . . ,Mn)	constructor/destructor application
D ::=	expressions
fail	failure
P, Q ::=	processes
out(N,M); P	output
in(N, x : T); P	input
!P	!P replication
0	nil
P | Q	parallel composition
new a : T; P	restriction
let x : T = D in P else Q	expression evaluation
if M then P else Q	conditional

6.2. Formal Verification Using Proverif

The modeling of a protocol in ProVerif is composed of declaration, process macros and main processes with queries to verify the security properties of a protocol. The ProVerif code is used to specify the protocol concisely using declaration of types, functions, queries and events. Free names are globally known, whereas bound names are locally known by the process such as the public channel for communication, while [private] excludes names from the attacker [58].

Specification includes the following:

- functions: `fun encrypt(bitstring, pkey): bitstring.`, key: `type key.`, private and public names: `free rand_a: bitstring[private] free pubChannel : channel.`
- Queries: On secrecy, reachability and authentication. A secrecy property is specified as a query of the attacker's knowledge `attacker(M)`. When the fact `attacker(M)` is derivable, the attacker may have the knowledge of `M`, but, if it is not derived, there is no way that the attacker can gain the knowledge of `M`. With reachability, the query `attacker(K)` is also used to debug the model of the protocol to check a particular branch is reachable or not, `query k: bitstring; event(endServer(k))`. The authentication properties are specified as correspondence assertions in the form of `event(e1(M)) event(e2(M))` which have to be proven. In the case of the DDSec and DDACap protocols, the following is queried to test the secrecy of message attributes such as `query attacker (rand_a)` for a nonce, `query attacker (ua)` for id and `query attacker (skuea)` for key. `query C:host,B:host,K:pkey, rand_rc:nonce; event(endUEB(B,C,K,rand_rc)) ==> event(beginUEA(B, C, K, rand_rc))` is used to test the events' relationship.
- Events: Querying events to test their relationship. (i) Event correspondence tested query is a basic correspondence assertion, `query x1:t1 . . , xn:tn; event(e(M1, ..,Mj)) ==> event (e'(N1,..,Nk))`, for the non-injective correspondence there is at least one earlier occurrence of `e'`. (ii) Injective correspondence assertions capture the one-to-one relationship, `query x1:t1, . ., xn:tn; inj-event(e(M1, . . , Mj)) ==> inj-event(e'(N1,..,Nk))`. The correspondence asserts that, for each occurrence of `event e(M1, . . ,Mj)`, there is a distinct earlier occurrence of `event e'(N1, . . ,Nk)`.
- Process: The protocol is encoded using main process and process macros for the participating entities to allow sub-process being defined as `(!process)`. The main process also starts off several copies of the system entities (UEs and gNB) with the relevant parameters representing several sessions of the roles as explained in the message exchange in Sections 5.1 and 5.2.

6.3. Formal Analysis of DDSec Protocol

The protocol is simulated using insecure public channel and processes representing the entities `(!processUEA())` for UE_A, `(!processUEB())` for UE_B and `(!processgNB())` for gNB. The security properties we are interested are secrecy, mutual authentication, authorization and secrecy on communication between the entities. In consideration with adversary, we formally analyze the protocol, and there were attacks on the protocol, as shown in Figure 6; hence, the protocol is not secure. ProVerif results show that the secrecy query holds `rand_b`, `UA, UB`, `d1`, `Acap` and `skuea, skueb` does hold, but secrecy of `rand_a` does not. In addition, the authentication of UE_A to UE_B does hold that is non-injective and injective agreements, but UE_B to UE_A does not hold. Thus, UE_B may end the protocol thinking it is talking to UE_A, while UE_A never runs the protocol with UE_B.

Event `beginUEA` means that UE_A has completed the protocol and UE_A received Message 6 and sent Message 7 Event `beginUEB` means that UE_B sent Message 6. All the parameters of the protocol are taken as arguments by these events: `A, B, pkueb, rand_a, rand_b` must verify the ID and respond to a nonce with a nonce. If the arguments match, then data are sent, otherwise it sends authentication failure and terminates the communication. We would like to prove the correspondence for

UE_A and UE_B authentication; `event(endUEB(A, B, K, rand_a, rand_b)) ==> event (beginUEA(A, B, K, rand_a,rand_b))` and `inj-event (endUEB(A, B, K, rand_a, rand_b)) ==> inj-event(beginUEA(A, B, K, rand_a, rand_b))`.

The direct proof of this correspondence in ProVerif does not hold. We also try to prove and conclude the desired correspondence by noticing that event which has `ub, d1, rand_a, rand_b, Ts_1` as argument cannot be executed before `ua, d1, rand_a` has been sent. That is, before `rand_a` request is has been executed with `rand_a, rand_b` response and which does not hold.

```
ededris@ededris-VirtualBox:~/proverif2.00$ ./proverif protocols/DDSec-attack.pv |grep RES
RESULT not attacker(rand_a[]) is false.
RESULT not attacker(rand_b[]) is true.
RESULT not attacker(skuea[]) is true.
RESULT not attacker(skueb[]) is true.
RESULT not attacker(ua[]) is true.
RESULT not attacker(ub[]) is true.
RESULT not attacker(d1[]) is true.
RESULT not attacker(Acap[]) is true.
RESULT event(endUEB(A_109,B_110,K,rand_a_111,rand_b_112)) ==> event(beginUEA(A_109,B_110,K
,rand_a_111,rand_b_112)) is false.
RESULT event(endUEA(A_113,B_114,K_115,rand_a_116,rand_b_117)) ==> event(beginUEB(A_113,B_1
14,K_115,rand_a_116,rand_b_117)) is true.
RESULT inj-event(endUEB(A_118,B_119,K_120,rand_a_121,rand_b_122)) ==> inj-event(beginUEA(A
_118,B_119,K_120,rand_a_121,rand_b_122)) is false.
RESULT (even event(endUEB(A_10519,B_10520,K_10521,rand_a_10522,rand_b_10523)) ==> event(be
ginUEA(A_10519,B_10520,K_10521,rand_a_10522,rand_b_10523)) is false.)
RESULT inj-event(endUEA(A_123,B_124,K_125,rand_a_126,rand_b_127)) ==> inj-event(beginUEB(A
_123,B_124,K_125,rand_a_126,rand_b_127)) is true.
```

Figure 6. DDSec protocol attack results.

6.3.1. The Attack on DDSec Protocol

The attack on a protocol can be explained using the attack derivation (abstracts) or attack trace (semantics). In ProVerif, the derivation explains the actions made by an attacker to break the security properties of a protocol, using abbreviations of internal representation of names or terms. It is a numbered list of steps each corresponding to a process or the attacker's action. While the attack trace represents the real attack as an executable trace of the considered process. A trace is a sequence of input and outputs on the public channel and of events in relation with the process. The input, output or event are followed by their location in the process at $\{n\}$, referring to the program point at the beginning of the process. As mentioned above, when ProVerif is given `query attacker (M)` where `M` is the message transmitted on the channel `c`, it intends to prove that a state in which a property is unreachable by showing `not attacker (M)`, the `RESULT not attacker (M)` is true, whereby the attacker does not have some term `(M)` i.e., message `M`. `query x1 : t1, . . , xn : tn ; event (e(M1, . . ,Mj)) ==> event (e'(N1, . . ,Nk))` tests the relationship between events. The correspondence asserts that, for each occurrence of the event, there is a distinct earlier occurrence of another event and if it does not then it is false.

Three attacks were found: the attacker I starts by eavesdropping on the communication between entities, impersonates UE_A continuing the protocol with UE_B, which completes the protocol with the attacker instead of UE_A as below.

```
UEA -> gNB: {AdvMsg}, {KgNBa}
UEB -> gNB: {IntMsg}, {KgNBb}
gNB -> UEA: {DiscMsg}, {KgNBa}
gNB -> UEA: {LinkUpMsg}, {KgNBb}
UEA -> I_UEB: {PublMsg}, {PKI}
I_UEA -> UEB: {PublMsg, {PKUEB}
UEB -> I_UEA: {LookUp}, {PKI}
I_UEB -> UEA: {LookUp}, {PKUEA}
UEA -> I_UEB: {SenData}, {PKI}
I_UEA -> UEB: {SenData}, {PKUEB}
```

6.3.2. Attack Derivation and Trace

TRACE 1

```
-- Query not attacker(rand_a[])
Completing...
Starting query not attacker(rand_a[])
goal reachable: attacker(rand_a[])

1 new skuea_36: skey creating skuea_2040 at {1}
2 new skua: sskey creating skua_2041 at {2}
3 new skueb_37: skey creating skueb_2042 at {3}
4 new skub: sskey creating skub_2043 at {4}
5 new kgnba: key creating kgnba_2044 at {5}
6 new kgnbb: key creating kgnbb_2045 at {6}
7 out(c, M) with M = pk(skuea_2040) at {8}
8 out(c, M_2190) with M_2190 = spk(skua_2041) at {10}
9 out(c, M_2262) with M_2262 = pk(skueb_2042) at {12}
10 out(c, M_2333) with M_2333 = spk(skub_2043) at {14}
11 insert keys(A,pk(skuea_2040)) at {15}
12 insert keys(B,pk(skueb_2042)) at {16}
13 The attacker has the message rand_a.
14 A trace has been found.
15 RESULT not attacker(rand_a[]) is false.
```

TRACE 2

```
16 new ua_66: id creating ua_7831 at {57} in copy a_7503
17 new ub_67: id creating ub_7832 at {58} in copy a_7503
18 new dataname_68: bitstring creating dataname_7833 at {59} in copy
a_7503
19 out(c, M_7890) with M_7890 = sencrypt((ub_7832,dataname_7833,
pk(skueb_7504)),kgnbb_7509) at {62} in copy a_7503
20 in(c, (PublMsg,a,a_7502)) at {63} in copy a_7503
21 event endUEB(A,B,pk(skueb_7504),rand_a,rand_b) at {64} in copy a_7503
(goal)
22 The event endUEB(A,B,pk(skueb_7504),rand_a,rand_b) is executed.
23 A trace has been found.
24 RESULT event(endUEB(A_107,B_108,K,rand_a_109,rand_b_110)) ==>
event(beginUEA(A_107,B_108,K,rand_a_109,rand_b_110)) is false.
```

The attacker's actions are explained in derivations and trace steps concisely, as shown in Figure 7; some text is omitted for simplicity. The three attack traces follow similar steps but have varying inputs, outputs and events as follows.

- The first trace, by eavesdropping the attacker knows `rand_a`, line 1–6 at inputs {1}–{6} corresponds to the creation of keys. Lines 7–10 at outputs {8}, {10}, {12}, {14} correspond to the public keys; they are stored in fresh variables `M`, `M_2190`, `M_2262`, `M_2333`, respectively, by the attacker for later reuse. Lines 11–12 at inputs {15}, {16} show the public keys are inserted in the key table in the main process for UE_A and UE_B. Then, the attacker gets `rand_a[]`.
- The second and third traces, Lines 1–12 at {1}–{16}, are the same as in Trace 1 while in Traces 2 and 3 the steps are the same but in different sessions `a_7503` and `a_9678`, respectively. With Lines 16–17 at inputs {57} and {58} corresponding to creations of `ua`, `ub`, the attacker creates `ua_7831`, `ub_7832` and stores them in `a_7503`. Thus, Line 19 at output {62} corresponds to a session `copy a_7503` UE_B has with the attacker, sending its ID `ub_7832` and data name `data name_7833`, which the attacker stores in a new variable `M_7890` for later use. In the same session, the attacker receives

(PublMsg,a,a_7502) at input {63}. Line 21 shows the attacker receiving his goal at {64} when event endUEB(A,B,pk(skueb_7504),rand_a,rand_b) is executed in a_7503 with UE_B as UE_B ends the protocol believing it was communicating with UE_A, but in fact it was communicating with the attacker. Trace 2 shows an attack on non-injective agreement, while Trace 3 shows an attack on injective agreement executed in session a_9678.

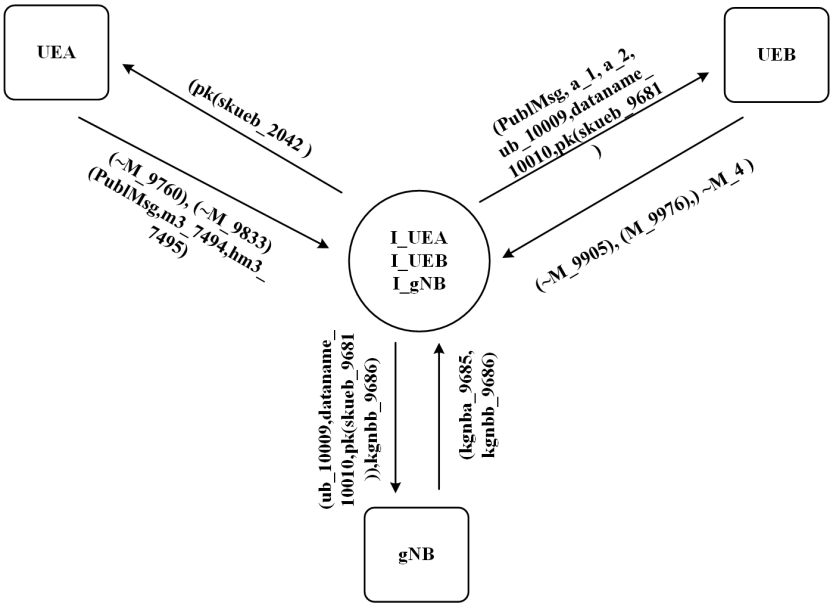

Figure 7. DDSec protocol attack trace.

6.4. Improved Version of DDSec Protocol

To address the attacks discovered in the previous version of DDSec protocol, changes were made in the protocol modeling. To improve the protocol, UE_A sends a nonce rand_na with the hash of its public key $hash(PK_{UE_A})$ to gNB in Message 1 and UE_B sends rand_nb with the hash of its public key $hash(PK_{UE_B})$ to gNB in Message 2. gNB returns rand_na in Message 3 and rand_nb to UE_B in Message 4. That prevents the attacker from starting a session with UE_B as it will have to get nonce rand_nb, ID UB and the right hash for the public key to be trusted by UE_B. The nonces together with the hashes of the public key provide the replay attack and integrity protection for Messages 1–4. The hashing of the keys adds another cryptographic operation to the protocol running as public keys are only known to the owner and gNB reducing the risk of being altered or leaked. The nonces sent in Messages 3 and 4 by gNB give assurance to UEA that UEB is trusted and vice versa. After the changes were made, the improved version of the protocol was run again and formally verified, as shown in Figure 8. ProVerif found no attack this time, hence the protocol is secure. ProVerif results indicate that the secrecy of rand_a, rand_b, d1, Acap, UA, UB, skuea, skueb holds and mutual authentication between UE_A and UE_B also holds as desired on injective and non-injective agreements.

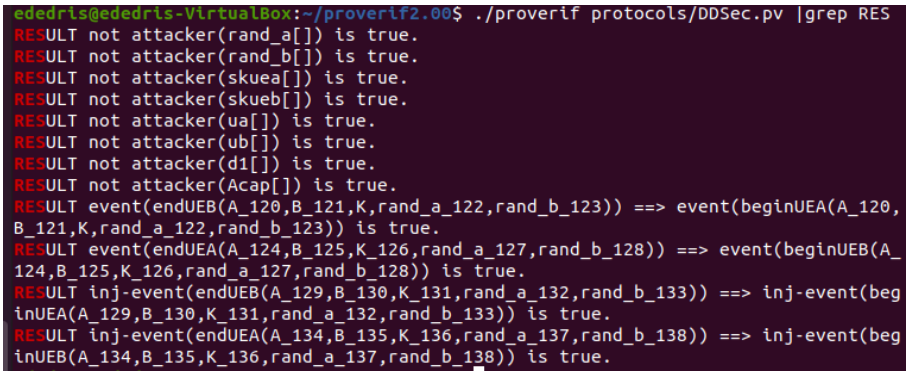

Figure 8. DDSec protocol safe results.

6.5. Formal Analysis of DDACap Protocol

The DDACap protocol was also simulated in ProVerif using insecure public channel and processes `(!ProcessUEB())` for UE_B and `(!ProcessUEC())` for UE_C. There was no attack found, as the ProVerif results show in Figure 9. The desired security properties are similar to those of the DDSec protocol; however, the IDs (UB/UC) are not queried for secrecy as they are sent in clear text in DDACap protocol. ProVerif results indicate that the secrecy of `rand_bc, rand_rc, skueb, skuec, d1, Acap` holds and mutual authentication of UE_B to UE_C holds in the form of non-injective and injective agreements, hence the protocol has no flaw.

The events that prove the correspondence and arguments taken, ID, $PK(X)$ and nonce are the same as DDSec protocol. If the arguments match, UE_B sends the data to UE_C, otherwise it sends authentication failure and terminates the communication. The correspondence proved in with the same methods used in Section 6.3. The direct proof of this correspondence in ProVerif holds because Message 3 is sent before Message 4. The correspondence is proved and concluded the desired correspondence by noticing that the event which has `uc, d1, rand_rb,rand_rc, Ts_4` as argument cannot be executed before `ub, d1, rand_rb` has been sent, that is, before `rand_rb` request has been executed with `rand_rb,rand_rc` response as part of the mutual authentication exchange. Which holds in ProVerif.

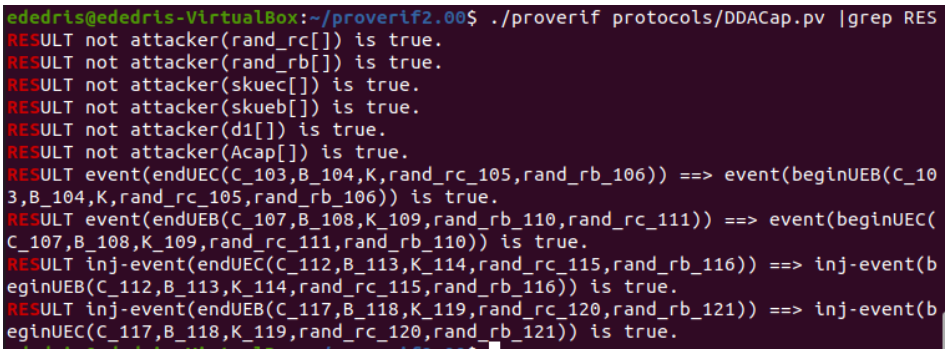

Figure 9. DDACap protocol safe results.

7. Security Analysis

This section presents the security analysis of the protocols' security properties and discusses the security consideration of the protocols. Since both DDSec and DDACap protocols have similar requirement and properties, this analysis focuses on the DDSec protocol.

7.1. Protocol Security Analysis

Our threat model assumes a Dolev–Yao adversary model [59]; it controls the network and can read, intercept, modify and send messages. It is also capable of carrying

out eavesdropping, manipulation, interception, impersonation and injection of messages attacks. The adversary can also apply hashing, encryption and sign on values that are known to the attacker. The analysis is based on the symbolic protocol model, assuming that the cryptography is perfect, and the computational strengths of the primitives are not considered. However, the protocol should meet certain security properties and the analysis of the protocols is based on security requirements in taxonomies set 1 [54] and set 2 [55], discussed as follows.

7.1.1. The Analysis Using Security Properties of Set 1

- Secrecy: This is achieved since `rand_a` and `rand_b` are never revealed to the attacker. It also covers confidentiality and privacy of the protocol's data.
- Aliveness: UE_A and UE_B obtain the aliveness of each other during $PublMsg$ and $LookUp$ messages exchange. UE_A gets non-injective agreement on `rand_a` with UE_B and UE_B with UE_A on `rand_b`.
- Weak Agreement: This is achieved when both entities obtain non-injective agreement on `rand_a, rand_b`, respectively, as UE_B ends running the protocol with UE_A guaranteed that UE_B has been talking to UE_B.
- Non-injective Agreement: UE_A obtains non-injective agreement on PK_{UE_A} and PK_{UE_B} with UE_B, respectively, and holds in ProVerif with `event(endUEB(A_107, B_108, K, rand_a_109, rand_b_110)) ==> event(beginUEA(A_107, B_108, K, rand_a_109, rand_b_110))`.
- Injective Agreement: It is central to the protocol's purpose; injective agreement between UE_A and UE_B on PK_{UE_A} and PK_{UE_B} assures each other that they are known. UE_A to UE_B and UE_B to UE_A holds with `inj-event (endUEB(A_116, B_117, K_118, rand_a_119, rand_b_120)) ==> inj-event (beginUEA(A_116, B_117, K_118, rand_a_119, rand_b_120))`.

7.1.2. The Analysis Using Security Properties of Set 2

- Mutual Entity Authentication: UE_A and UE_B authenticated each other with `A, B, pkuea, rand_a, rand_b` when they proved to hold, hence enforcing this requirement.
- Mutual Key Authentication: The public keys PK_{UE_A} and PK_{UE_B} are self-certified, used as identifiers and hashed. This implicitly authenticates the public keys.
- Mutual Key Confirmation: This requirement does not apply to this protocol as the UEs generate their own public and private keys. However, since they both use ECIES that partly enforces this agreement, the successful authentication of UE_A to UE_B and UE_B to UE_A implicitly enforces this requirement.
- Key Freshness: ProVerif has no function to check key freshness, however the messages exchange includes nonces and timestamps, hence checking the freshness of messages which includes the keys. Since the secrecy of these keys are not violated, it implies keys are fresh.
- Unknown-Key Share: Since the entities use their own unique key pair and private keys SK_{UE_i} are not violated, it enforces this requirement. The entities' ID and generated hashes prevent this attack. The inclusion of UA and UB in the authentication process also proves this requirement.
- Key Compromise Impersonation Resilience: Since the entities use their own unique public/private keys using ECIES, it enforces this requirement. Therefore, knowing one key in a session is not enough to deduce another. Backward and forward secrecy of keys are possible; no entity or adversary is capable of computing keys in past session or predict feature keys, which is due to ECC. Obviously, the public keys are globally known but the private keys are only known to the owners. However, to compromise the keys, ECIES, UE_A and UE_B would have to be compromised at the same time. It should be noted that compromising the HN during the primary authentication does not necessarily mean that D2D authentication will be compromised.

The analysis of the DDACap protocol is similar to that of DDSec, the difference being the parameters such as IDs, tokens, nonce, hashes and cryptographic primitives used and the exchanged messages, but the security requirements and security properties are the same.

7.2. Security Consideration

The UEs using the proposed D2D security protocols will be able to authenticate each other, share data and delegate their access rights via security tokens in network- and non-network-assisted D2D communications scenarios. The participating entities use ECC, whereby solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) is not feasible. Hence, attackers cannot break the proposed protocols' cryptography, Key Derivation Function (KDF) and find the secret key if a big size of the binary elliptic curve is used as it requires very high computational time. The larger is the curve size, the better is the security level of a protocol. In addition, the use of randomness, hash function, tokens, delegation, ID and certificate digital signature together with Third Generation Partnership Project (3GPP)'s 5G security standard recommendations harden the security of the protocols.

The formal analysis proves that the DDSec and DDACap protocols can address 5G-enabled D2D communications security issues as proposed by providing entity authentication, authorization, decentralization, anonymity, scalability, fault tolerance, user permission delegation, secrecy, data authenticity and integrity during D2D data sharing.

8. Conclusions

In this paper, we discuss how 5G will use D2D communications to improve quality of experience and service for end users and enable MNO to offload traffic from the backhaul and push content to the edge. This will enable mobile users with their UE to access, cache and share ProSe and edge content. We explore the related work on 5G-enabled D2D communications and end users' role in content distribution and the integration of D2D communications with CCN to solve the issues of service delivery and traffic offloading. However, such integrated system lack an efficient and robust multi-purpose security solution that addresses its security issues. Thus, we propose a D2D security solution that applies IBE, ECIES and fine-grained access controls for authentication and authorization procedures. The proposed solution consists of DDSec and DDACap protocols that can be applied to enable D2D users to cache, share data and delegate their access rights to other D2D users in different coverage scenarios with or without network assistance. The proposed protocols were formally verified using formal methods and automated protocol verifier ProVerif. We also comprehensively analyzed the protocols' security properties with two taxonomies. The future work should be on how this solution can be extended to V2V and IoT.

Author Contributions: Conceptualization, E.K.K.E., M.A. and J.L.; methodology, E.K.K.E. and M.A.; validation, E.K.K.E. and M.A.; formal analysis, E.K.K.E.; investigation, E.K.K.E.; writing—original draft preparation, E.K.K.E.; writing—review and editing, E.K.K.E. and M.A.; and supervision, M.A. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. 3GPP. Security architecture, procedures for 5G system. In *Technical Specification (TS) 3GPP TS 33.501 V17.0.0 (2020–2012)*; Third Generation Partnership Project; 3GPP: Sophia Antipolis, France, 2020.
2. Edris, E.K.K.; Aiash, M.; Loo, J. Investigating Network Services Abstraction in 5G enabled Device-to-Device (D2D) Communications. In *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*; IEEE: Leicester, UK, 2019; pp. 1660–1665. [CrossRef]
3. Yu, H.; Afzal, M.K.; Zikria, Y.B.; Rachedi, A.; Fitzek, F.H.P. Tactile Internet: Technologies, test platforms, trials, and applications. *Future Gener. Comput. Syst.* **2020**, *106*, 685–688. [CrossRef]

4. Feng, D.; Lai, L.; Luo, J.; Zhong, Y.; Zheng, C.; Ying, K. Ultra-reliable and low-latency communications: Applications, opportunities and challenges. *Sci. China Inf. Sci.* **2021**, *64*, 1–12. [CrossRef]
5. Singh, K.; Ku, M.L.; Flanagan, M.F. Energy-Efficient Precoder Design for Downlink Multi-User MISO Networks With Finite Blocklength Codes. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 160–173. [CrossRef]
6. Liu, L.; Yu, W. A D2D-based Protocol for Ultra-Reliable Wireless Communications for Industrial Automation. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 5045–5058. [CrossRef]
7. 3GPP. Study on architecture enhancements to support Proximity-based Services (ProSe). In *Technical Specification (TS) 3GPP TR 23.703 V12.0.0 (2014-02)*; Third Generation Partnership Project; 3GPP: Sophia Antipolis, France, 2014.
8. Gupta, A.; Jha, R.K. A Survey of 5G Network: Architecture and Emerging Technologies. *IEEE Access* **2015**, *3*, 1206–1232. [CrossRef]
9. Chandrasekaran, G.; Wang, N.; Hassanpour, M.; Xu, M.; Tafazolli, R. Mobility as a Service (MaaS): A D2D-Based Information Centric Network Architecture for Edge-Controlled Content Distribution. *IEEE Access* **2018**, *6*, 2110–2129. [CrossRef]
10. Edris, E.K.K.; Aiash, M.; Loo, J. The Case for Federated Identity Management in 5G Communications. In Proceedings of the 5th IEEE International Conference on Fog and Mobile Edge Computing (FMEC 2020), Paris, France, 20–23 April 2020; IEEE: Paris, France, 2020.
11. 5GPPP. *Deliverable D2.7 Security Architecture (Final)*; Technical Report, 5G Enablers for Network; 5GPPP: Heidelberg, Germany, 2017.
12. 3GPP. 5G system stage 2 Rel-17. In *Technical Report 3GPP TSG Rel-17 (2021)*; Third Generation Partnership Project; 3GPP: Sophia Antipolis, France, 2021.
13. 3GPP. Proximity-based services (ProSe) Stage 2. In *Technical Specification (TS) 3GPP TS 23.303 V16.0.0 (2020-07)*; Third Generation Partnership Project; 3GPP: Sophia Antipolis, France, 2020.
14. Qiao, J.; Shen, X.; Mark, J.; Shen, Q.; He, Y.; Lei, L. Enabling device-to-device communications in millimeter-wave 5G cellular networks. *IEEE Commun. Mag.* **2015**, *53*, 209–215. [CrossRef]
15. Zhang, T.; Fang, X.; Liu, Y.; Nallanathan, A. Content-centric mobile edge caching. *IEEE Access* **2019**, *8*, 11722–11731. [CrossRef]
16. Zhang, A.; Chen, J.; Hu, R.Q.; Qian, Y. SeDS: Secure Data Sharing Strategy for D2D Communication in LTE-Advanced Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 2659–2672. [CrossRef]
17. Zhang, A.; Lin, X. Security-Aware and Privacy-Preserving D2D Communications in 5G. *Netw. IEEE* **2017**, *31*, 70–77. [CrossRef]
18. Melki, R.; Noura, H.N.; Chehab, A. Lightweight and Secure D2D Authentication & Key Management Based on PLS. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–7. [CrossRef]
19. Cao, M.; Chen, D.; Yuan, Z.; Qin, Z.; Lou, C. A lightweight key distribution scheme for secure D2D communication. In Proceedings of the 2018 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT), Tangier, Morocco, 20–22 June 2018; pp. 1–8. [CrossRef]
20. Wang, M.; Yan, Z.; Niemi, V. UAKA-D2D: Universal Authentication and Key Agreement Protocol in D2D Communications. *Mob. Netw. Appl.* **2017**, *22*, 510. [CrossRef]
21. Wang, M.; Yan, Z.; Song, B.; Atiquzzaman, M. AAKA-D2D: Anonymous Authentication and Key Agreement Protocol in D2D Communications. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 19–23 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1356–1362. [CrossRef]
22. Gope, P. LAAP: Lightweight anonymous authentication protocol for D2D-Aided fog computing paradigm. *Comput. Secur.* **2019**, *86*, 223–237. [CrossRef]
23. Seok, B.; Sicato, J.C.S.; Erzhena, T.; Xuan, C.; Pan, Y.; Park, J.H. Secure D2D communication for 5G IoT network based on lightweight cryptography. *Appl. Sci.* **2020**, *10*, 217. [CrossRef]
24. Wang, P.; Chen, C.M.; Kumari, S.; Shojafar, M.; Tafazolli, R.; Liu, Y.N. HDMA: Hybrid D2D message authentication scheme for 5G-enabled vanets. *IEEE Trans. Intell. Transp. Syst.* **2020**. [CrossRef]
25. Lopes, A.P.G.; Gondim, P.R. Mutual authentication protocol for D2D communications in a cloud-based e-health system. *Sensors* **2020**, *20*, 2072. [CrossRef]
26. Wang, L.; Tian, Y.; Zhang, D.; Lu, Y. Constant-round authenticated and dynamic group key agreement protocol for D2D group communications. *Inf. Sci.* **2019**, *503*, 61–71. [CrossRef]
27. Wang, L.; Liu, J.; Chen, M.; Gui, G.; Sari, H. Optimization-based access assignment scheme for physical-layer security in D2D communications underlaying a cellular network. *IEEE Trans. Veh. Technol.* **2018**, *67*, 5766–5777. [CrossRef]
28. Li, Z.; Hu, H.; Hu, H.; Huang, B.; Ge, J.; Chang, V. Security and Energy-aware Collaborative Task Offloading in D2D communication. *Future Gener. Comput. Syst.* **2021**, *118*, 358–373. [CrossRef]
29. Yan, Z.; Xie, H.; Zhang, P.; Gupta, B.B. Flexible data access control in D2D communications. *Future Gener. Comput. Syst. Future Gener. Comput. Syst.* **2018**, *82*, 738–751. [CrossRef]
30. Li, Q.; Huang, L.; Mo, R.; Huang, H.; Zhu, H. Robust and scalable data access control in D2D communications. *IEEE Access* **2018**, *6*, 58858–58867. [CrossRef]
31. Kang, H.J.; Kang, C.G. Mobile device-to-device (D2D) content delivery networking: A design and optimization framework. *J. Commun. Netw.* **2014**, *16*, 568–577. [CrossRef]

32. Golrezaei, N.; Molisch, A.F.; Dimakis, A.G.; Caire, G. Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution. *Commun. Mag. IEEE* **2013**, *51*, 142–149. [CrossRef]
33. Bastug, E.; Bennis, M.; Debbah, M. Living on the edge: The role of proactive caching in 5G wireless networks. *Commun. Mag. IEEE* **2014**, *52*, 82–89. [CrossRef]
34. Jacobson, V. A Description of Content-Centric Networking (CCN). In *Future Internet Summer School (FISS)*; 2009; Volume 2018. Available online: https://named-data.net/wp-content/uploads/2014/04/van-ccn-bremen-description.pdf (accessed on 2 July 2021).
35. Checko, A.; Christiansen, H.; Yan, Y.; Scolari, L.; Kardaras, G.; Berger, M.; Dittmann, L. Cloud RAN for Mobile Networks-A Technology Overview. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 405–426. [CrossRef]
36. 3GPP. Study on Architecture for Next Generation System. In *Technical Specification (TS) 3GPP TR 23.799 V14.0.0 (2016-12)*; Third Generation Partnership Project; 3GPP: Sophia Antipolis, France, 2016.
37. Ravindran, R. Enabling ICN in 3GPP's 5G NextGen Core Architecture. In *Memo ICNRG*; IETF: Fremont, CA, USA, 2019; Volume 2019.
38. Edris, E.K.K.; Aiash, M.; Loo, J.; Alhakeem, M.S. Formal Verification of Secondary Authentication Protocol for 5G Secondary Authentication. *Int. J. Secur. Netw.* **2020**, in press.
39. Edris, E.K.K.; Aiash, M.; Loo, J. Network Service Federated Identity (NS-FId) Protocol for Service Authorization in 5G Network. In Proceedings of the 5th IEEE International Conference on Fog and Mobile Edge Computing (FMEC 2020), Paris, France, 20–23 April 2020; IEEE: Piscataway, NJ, USA, 2020.
40. Altmann, V.; Skodzik, J.; Danielis, P.; Mueller, J.; Golatowski, F.; Timmermann, D. A DHT-Based Scalable Approach for Device and Service Discovery. In Proceedings of the 12th IEEE International Conference on Embedded and Ubiquitous Computing, Milan, Italy, 26–28 August 2014; pp. 97–103.10.1109/EUC.2014.23. [CrossRef]
41. 3GPP. Feasibility study on the security aspects of remote provisioning, change of subscription for Machine to Machine (M2M) equipment. In *Technical Specification (TS) 3GPP TR 33.812 V9.2.0 (2010-06)*; Third Generation Partnership Project; 3GPP: Sophia Antipolis, France, 2010.
42. Wang, K.; Yu, F.R.; Li, H.; Li, Z. Information-Centric Wireless Networks with Virtualization and D2D Communications. *IEEE Wirel. Commun.* **2017**, *24*, 104–111. [CrossRef]
43. Gandotra, P.; Jha, R.K.; Jain, S. A survey on device-to-device (D2D) communication: Architecture and security issues. *J. Netw. Comput. Appl.* **2017**, *78*, 9–29. [CrossRef]
44. Haus, M.; Waqas, M.; Ding, A.Y.; Li, Y.; Tarkoma, S.; Ott, J. Security and Privacy in Device-to-Device (D2D) Communication: A Review. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1054–1079. [CrossRef]
45. Nunes, I.O.; Tsudik, G. KRB-CCN: Lightweight Authentication & Access Control for Private Content-Centric Networks. In Proceedings of the International Conference on Applied Cryptography and Network Security, Leuven, Belgium, 2–4 July 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 598–615.
46. Tourani, R.; Misra, S.; Mick, T.; Panwar, G. Security, Privacy, and Access Control in Information-Centric Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 566–600. [CrossRef]
47. Alliance, N. 5G security recommendations Package #2: Network Slicing. In *White Paper*; NGMN: Frankfurt am Main, Germany, 2016; pp. 1–12.
48. Aamir, M.; Zaidi, S.M.A. Denial-of-service in content centric (named data) networking: A tutorial and state-of-the-art survey. *Secur. Commun. Netw.* **2015**, *8*, 2037–2059. [CrossRef]
49. Lichtman, M.; Rao, R.; Marojevic, V.; Reed, J.; Jover, R.P. 5G NR Jamming, Spoofing, and Sniffing: Threat Assessment and Mitigation. In Proceedings of the 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, USA, 20–24 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
50. Edris, E.K.K.; Aiash, M.; Loo, J. Formal Verification and Analysis of Primary Authentication based on 5G-AKA Protocol. In *The Third International Symposium on 5G Emerging Technologies (5GET 2020)*; IEEE: Paris, France, 2020.
51. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. *SIAM J. Comput.* **2003**, *32*, 586–615. doi:10.1137/S0097539701398521. [CrossRef]
52. SECG. *SEC 1: Recommended Elliptic Curve Cryptography*; SECG: Guangzhou, China, 2009.
53. Girault, M. Self-certified public keys. In *Workshop on the Theory and Application of of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1991; pp. 490–497. [CrossRef]
54. Lowe, G. A hierarchy of authentication specifications. In Proceedings of the 10th Computer Security Foundations Workshop, Rockport, MA, USA, 10–12 June 1997; IEEE: Piscataway, NJ, USA, 1997; pp. 31–43. [CrossRef]
55. Menezes, A.J.; Oorschot, P.C.V.; Vanstone, S.A. *Handbook of Applied Cryptography*; Includes Bibliographical References and Index, ID: alma991001301199704781; CRC Press: Boca Raton, FL, USA, 2018.
56. Aiash, M.; Loo, J. A formally verified access control mechanism for information centric networks. In Proceedings of the 12th International Joint Conference on e-Business and Telecommunications (ICETE), Colmar, France, 20–22 July 2015; IEEE: Piscataway, NJ, USA, 2015; Volume 4, pp. 377–383.
57. Armando, A.; Basin, D.A.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuellar, J.R.; Drielsma, P.H.; Heam, P.C.; Kouchnarenko, O.; Mantovani, J.; et al. The AVISPA tool for the automated validation of Internet security protocols and applications. *Comput. Aided Verif. Proc.* **2005**, *3576*, 281–285.

58. Blanchet, B.; Smyth, B.; Cheval, V.; Sylvestre, M. ProVerif 2.01: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial. 2020. Available online: https://opam.ocaml.org/packages/proverif/ (accessed on 2 July 2021).
59. Dolev, D.; Yao, A.C.C. On the Security of Public Key Protocols. *IEEE Trans. Inf. Theory* **1983**, *30*, 198–208. [CrossRef]
60. Ryan, M.D.; Smyth, B. Applied pi calculus. *Form. Model. Tech. Anal. Secur. Protoc.* **2011**, *5*, 112–142.